The I/O Connector

The Newsletter of the San Diego Atari Computer Enthusiasts

SAN DIEGO ATARI COMPUTER ENTHUSIASTS

is an independent, non-profit organization and user group
with no connection to the ATARI Corporation. Membership fees
are currently $15.00 annually, from January 1 thru December
31 of the current calendar year. Membership includes free
acess to the computer program library, subscription to the
"I/O Connector", and classes when held. Permission to reprint
articles in any non-commercial form is permitted with
specific authorization, as long as proper credit is given.

COMMERCIAL ADVERTISING RATES
FULL PAGE — $40.00; HALF PAGE — $20.00
QUARTER PAGE — $10.00; BUSINESS CARD $5.00

S.D.A.C.E. OFFICERS

President	Buck Bragunier	465-0831
V. President	Rick DeHaven	284-2365
Treasurer	Norm Davis	448-2971
Secretary	George Hancher	238-9048
Program Chairman	Dick Hiatt	463-8460
Membership	Norm Davis	448-2971
Newsletter	David Delgadillo	262-4829

Librarians:

Santee	Tim Fraser	448-8231
North County	Jim Gollwitzer	486-2479
Ocean Beach	Frank Miller	223-6378
South Bay	Jesse Adame	475-3524
Tierra Santa	Peter Payne	560-4272

(Calls between 5:00 AND 9:00, PLEASE)

CORRESPONDENCE ADDRESS
SAN DIEGO ATARI COMPUTER
ENTHUSIASTS
P.O. Box 203076
San Diego, Ca. 92120

BULLETIN BOARD
S.D.A.C.E. BBS
Sysop: Mark Booth
(619) 462-6302
24 Hours

SUBMISSIONS TO THE NEWSLETTER
are most welcome, and due by the 28th of the month for
publication in the next month's newsletter. Mail
double-spaced text or (returnable) disks with text files to
the Editor.

BUY/SELL/TRADE
ads are free to members of S.D.A.C.E. members. Ads must be 25
words or less, typewritten copy, please. The Editor will
accept ads at the meetings or those mailed in to the above
correspondence address. Deadline for ads is the same for
articles — 28th of the month.

SAN DIEGO BBS LIST

```
Alice's Restuarant......................585-1397 *
Arena, The..............................753-1478
Atari Line..............................282-6608 +
BBS, The................................447-8143
Bill Board, The.........................282-5219
Dog House, The..........................582-7447
Game Line...............................270-0111
Poop Deck, The..........................589-0565
Rick's Place............................284-3821
Rorinar Valley..........................729-2898
S.D.A.C.E. BBS..........................462-6302
S.D. Air Traffic Control................588-4682
San Diego BBCS..........................483-0495 #
Savage's Hut............................466-4238
Sherwood Forest.........................483-2592
Stallion's Corner.......................428-0319
U.F.P. Starship.........................434-6270
WareHouse, The..........................757-4135
```
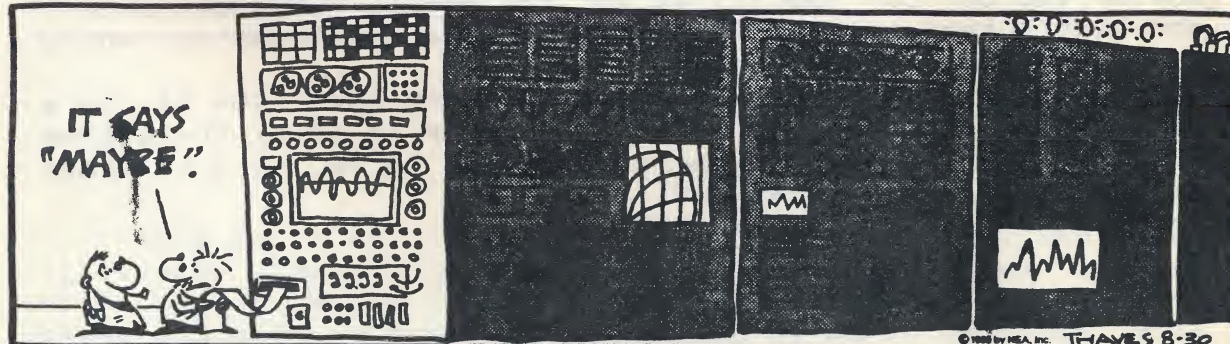
The above is a current list of Atari oriented BBS' in San Diego. Many of them are 300/1200 baud and open 24 hours. S.D. Air Traffic Control is planning on going 2400 baud very soon. This may not be a complete list at this time. We will reprint an updated list in January 1986. Please let the newsletter editor know of any additions or changes. Next month we will have a national Atari BBS list in a handy pull out section.

NOTES:
  * Sign on in ASCII, home of Dimension-X
  + Limited hours, evenings and weekends
  # Sign on in ASCII

FRANK AND ERNEST   **Bob Thaves**

Mike Smith wrote a letter to the president of Electronic Arts protesting the fact that they were not making Skyfox for the Atari computer. This letter contains the reply that Mike received from Trip Hawkins (EA President) on September 7,1985.

August 29,1985

Dear Mr. Smith:

Thanks for your interest in Electronic Arts products and your concern over the lack of new Atari titles. I'm unhappy about the situation, too, but let me explain our position.

In spite of the number of Ataris that have been reported sold to consumers, the number that has the most meaning for us is the number of pieces of Atari software that we have sold. That number is small -- too small to justify the expense of developing new titles for the machine. Apparently, many Atari owners don't necessarily buy much software or even use their macines. The Apple, Commodore, and IBM version of our titles far outsell the Atari versions. Some of these titles were introduced on Atari first but didn't sell well until they were translated for another machine.

We'd like to keep all our customers happy, but we need to make enough money to stay in business. It's not good business to develop an Atari title when too few will sell to recover our costs. We are one of the few software companies making a profit right now, but we would not be profitable if we had depended on Atari software sales.

Our artists are also trying to make money. They want to develop products that will sell well and earn them large royalties. They may prefer to program on an Atari -- but not if it means a much smaller income.

We haven't given up entirely. We have plans to translate one or two of our new titles to the Atari. We're cautiously optimistic that they will sell well. One of those currently under development is Skyfox.

We hope Ataris continue to sell and that the ST is a success. But no matter how much hardware is sold, the bottom line for us is software sales -- that is up to all Atari owners, not just a loyal few.

Thanks again for your interest.

Sincerely,

Trip Hawkins
President

WHO IS TRIP HAWKINS ANYWAY?

From ABACUS
By Charles Cherry

Trip Hawkins, the President of Electronic Arts, has become the most quoted guru in the Silicon Valley. These days I see his name more often than Esther Dyson's and Phillipe Kahn's combined. His story is always the same, praising the Amiga and bashing the ST. What's with Trip? Why does he hate Atari so much?

Last fall he said he would stop supporting Atari computers because of all the pirating going on. There are undoubtedly Atari pirates, but what about the Apple pirates who trade commercial software openly on bulletin boards and have given all user groups a bad name? The announcement came about the time that other publishers were cutting back on Atari software because sales were slow and many thought Atari would disappear. I suspect pirating was not the real issue.

Now Trip's featured in full page ads extolling the Amiga. No software company has ever promoted one brand of computer before. Who's paying for those ads and why? Maybe Trip is just a True Believer. There are people who fell that way about the Macintosh. But it does make his credibilty as an impartial guru a little shaky.

Electronic Arts is not going to translate thier software to the ST (except the Financial Cookbook, which is all text). More importantly, they're not going to develope new software for the ST. With their history of excellent programs implemented on many computers, it is sad to see them hitch their wagon to one machine, especially one with so little chance of success.

The boon in sales of the XLs and XEs has caused Electronic Arts to rethink its policy on Atari 8-bit machines. Titles are slowly being translated. We hope that the success for the ST will encourage them to change their short-sighted 68000 policy.

DISK OF THE MONTH

This month's disk of the month contains AMS1 and AMS2 files with the appropriate player for each. Files include Hill: Street Blues theme, Ghostbusters theme, Pink Panther theme and many others. These disks, as well as those from past months, will be on sale for $5.00 at the SDACE meeting.

CHRIS CRAWFORD
ASSEMBLY LANGUAGE COURSE
FOR WORLDWIDE USERS NETWORK

LECTURE ONE - 5/28/85
WHY LEARN ASSEMBLY LANGUAGE?

Assembly language is the great barrier that divides the professional programmer from the amateur. It is the most powerful language available for a microcomputer.

There are four reasons for learning to program in assembly language. First, the speed of execution of assembly language is very high -- about ten times higher than BASIC on the average, perhaps a thousand times faster on certain operations. Even ACTION, the fastest high-level language, is only about half as fast as assembly language.

Second, assembly language tends to be more compact than many languages. Again, ACTION! provides a good comparison. Code produced by ACTION! is about twice as large as equivalent assembly language.

The third reason to program in assembly language is that assembly gives you access to features of the machine that simply are not available in high-level languages. Interrupts are the most notable examples.

Finally, the most important reason for learning to program in assembly language is that it will help you to understand the machine better. And that is a very good place to begin, for you cannot learn assembly language unless you know a little bit about computers.

HOW COMPUTERS WORK

I am now going to describe how computers work, in very rough terms. Computers operate on a hierarchy of concepts that spans a great range, rather like the hierarchy that starts with protons and electrons, moves through atoms, molecules, cells, people to civilizations.

A civilization is composed of protons and electrons, but to understand how it is so composed one must know a great deal about the intermediate steps. So too is a computer composed of transistors. There are four intermediate steps between the transistor and the computer.

A transistor is an electrically operated switch. We can assemble transistors into gates that will turn circuits on or off depending on the states of other circuits. There are a variety of gates reflecting the various Boolean operations:

AND, OR, NOT, NAND, NOR and EOR.

Gates can be assembled into latches, decoders, and adders. A latch is the simplest memory element: it remembers one bit of information. A decoder translates a number encoded in binary form on a few wires into a selection of one of many wires. An adder will add two one-bit values, with a carry, and generate a carry of its own.

We can next broaden each of these devices into an eight-bit device by simply slinging the devices side by side. Eight one-bit latches slung side-by-side gives one byte of RAM. Eight adders make an eight-bit adder.

We can thus create a RAM module by building many bytes of RAM. We access this RAM module with three buses: a data bus, an address bus, and a control bus. The data bus carries information between the central processing unit and the RAM module.

The address bus is sixteen bits wide; a decoder in the RAM module takes the numeric value on the address bus and decodes it to select the single byte of RAM that is indicated by the address. The control bus establishes the direction of the data flow on the data bus and the timing of data transfer.

The central processing unit (CPU) represents the highest intellectual level of the computer. It is composed of four parts: the Arithmetic and Logic Unit (ALU), the registers, the address bus controller, and the instruction decoder. The ALU is composed of adders and gate arrays that crunch numbers. The particular device to use is selected with a decoder.

The registers are simply on-board RAM. The address bus controller is a device that puts the desired RAM address onto the address bus. The real heart of the CPU is the instruction decoder, a very complex decoder that takes the program instructions out of RAM and translates them into action. It does this by feeding the instructions (which are numbers) into decoder circuits that activate the desired gateways in the CPU.

PROGRAMMING A MICROPROCESSOR

Machine code is nothing more than a bunch of numbers that mean something to the CPU. It's hard to work with pure numbers, so we use a little code that makes it easier for us to understand the codes that the computer uses. This programmer-friendlier code is called assembly language, It is a direct, one-to-one translation of machine code. Here is an example of the two side by side:

```
Machine Code      Assembly Language
A9   05 LDA       #FINGERS
133 . $9C         STA COUNT
```

The code on the right may not look very readable, but
you must agree, it's far more readable than the code on the
left. And they both mean exactly the same thing.

     Unfortunately, the computer cannot read the assembly
code, only the machine code. Therefore, we need a translator
program that will translate the easier-to-understand code on
the right into the impossible-to-understand code on the left.
This translator program is called an assembler.

     A program that goes in the reverse direction,
translating machine code to assembly, is called disassembler.
It may seem like a bother to go through all the hassle of
using an assembler, but it is actually much easier.

     Assembly language is not only more readable than machine
code, but it is also assembly-time relocatable; this means
you can move it around in RAM freely before you start the
assembly process. A good assembler also offers a number of
extra features that makes it easier to keep track of your
program or modify it quickly.

USING AN ASSEMBLER

     There are three steps involved in writing an assembly
language program: editing, assembling, and debugging. Editing
is the process of typing in your assembly language
statements. Assembling is the invocation of the assembler.
Debugging is the process of running your program and
analyzing why it doesn't work. Thus, the entire process of
writing an assembly-language process can be described by a
fictitious BASIC program:
```
 FOR I=1 to 1,000,000,000...
 EDIT PROGRAM
 ASSEMBLE PROGRAM
 DEBUG PROGRAM
 NEXT I
```

THE 6502 MICROPROCESSOR

     The first item in the 6502 that I will describe is the
accumulator. This is a single one-byte register in the 6502.
It is the central workbench of the microprocessor; almost
everything happens in the accumulator. Your first three
instructions on the 6502 are:
1. LDA address (Load the Accumulator with the contents of
 address)

     This instruction loads the accumulator with the contents

of the memory location specified by the value of address. The address can be specified by either an outright value, such as $0600, or a symbolic reference, such as FISH, where the value of FISH has been previously declared by, say, an ORG statement or an equate statement.

2, LDA #value (Load the Accumulator with value)

This is much like the earlier statment; it loads the accumulator with a number, only the number loaded is specified immediately rather than stored in a memory location. Thus, the command LDA # 9 will put a 9 into the accumulator.

3. STA address (Store the Accumulator into address)

This command will store the contents of the accumulator into the RAM location whose address is specified in the command. It is just like the first command, except that the direction of data motion is reversed. The LDA command is like a read, which the STA is like a write.

You are now equipped to move data around inside the computer. These commands will allow you to read data from one area of memory and store it into another. LDA and STA are the two most common instructions used in any 6502 program.

Exercise: Write a program that will read the contents of address $FE00 and store the result into addres $680. Your biggest problem here will be just getting your assembler to work. Therefore, I will give the answer away:

```
ROMADD ORG $FE00
RAMADD ORG $680
       ORG $600
       LDA ROMADD
       STA RAMADD
       BRK
       END
```

That's the program. Try to get it running with your assembler.

NOVEMBER ELECTIONS

November is coming and so are elections for club officers. I have taken those portions of our club Constitution and placed them on these pages for your perusal. Also included are the duties of each elected officer.

ARTICLE IV-OFFICERS

Section A — Titles
The officers of this organization shall be: President, Vice President, Secretary and Treasurer.

Section B — Elections
   1. Elections shall be held at the November meeting.
   2. New officers shall take office at the January meeting.

ARTICLE II-OFFICERS

Section A
Eligibility of officers:
   1. Any regular member for at least six months prior to the election.
   2. The member shall be in good standing.

Section B
Duties of officers:
   1. The President shall:
      A. Preside at all meetings of the organization and Board of directors.
      B. Appoint all committee chairmen.
      C. Be an ex-officio member of all committees.
      D. See that regulations of the organization are enforced and have general supervision of the work of the club.
      E. Carry out assignments and instructions given him/her by the vote of the organization.
      F. Perform all such other duties as customarily pertain to the office of the president.
      G. Appoint a member in good standing to fill vacancies which occur in elective offices for the remainder of the term, to be confirmed by a majority vote of the membership at the next meeting.

2. The Vice-President shall:
     A. Be an aide to the president.
     B. Preside over all meetings in the absence of the president.
     C. Co-ordinate all committees and be responsible for their proper functioning.

3. The Secretary shall keep a record of proceedings of all meetings; regular, Board of Directors and Special meetings.

4. The Treasurer shall:
     A. Collect all fees and dues.
     B. Receive and Disburse funds according to the By-Laws.
     C. Keep an accurate account of all organization funds.
     D. Co-sign all warrants with the president.
     E. Make a Financial report to the board of Directors at each meeting which shall show balance on hand at the last meeting, receipts and/or expenditures since that time and the balance on hand at the present time.
     F. Submit all books for audit at the end of December. The auditor shall report his findings of the said audit at the next meeting.

Section C Elections
   1. All officers shall be nominated by a nominating committee appointed by the president at least 2 weeks prior to the election. Further nominations may be made from the floor at the time of the election by any member in good standing.
   2. Election shall be by secret ballot.

Motorola's MC68000

by SIG-ATARI's Tim Barr

[Attention all you hackers out there! Are you anxious to get the new ST Line of Atari Computers? Here's a little background info about the heart and soul of these new computers. This article was taken from the SIG Atari section of Compuserve.]

The MC68000 is a 16-bit MPU with 17 general purpose 32-bit registers, a 32-bit program counter and a 16-bit status register. The first eight GP registers (D0-D7) are used as data registers. The next seven GP registers (A0-A6) and the system stack pointers can be utilized as address registers and pointers for software use. The data registers can be used for different data sizes. These sizes are: BYTE (8-bit), WORD (16-bit), and LONG WORD (32-bit) operations. The MPU has a 24-bit address bus (actually it is 23-bit address line and a HI BYTE and LOW BYTE select lines which gives you 8 megawords or 16 megabytes of memory) and a 16-bit data bus.

Five basic data types are supported by the instruction set. These data types are: BITS, BCD DIGITS (4-bits), BYTES (8-bits), WORDS (16-bits), and LONG WORDS (32-bits). The MPU has 14 address modes of six basic types: REGISTER DIRECT, REGISTER INDIRECT, ABSOLUTE, PROGRAM COUNTER RELATIVE, IMMEDIATE, AND IMPLIED. The surprising thing about the 68000 is that it only has 56 instruction types and a total of only 88 actual instructions. The actual 16-bit OP-CODE that the system uses is a combination of an instruction and an addressing mode, GP register number, an OP-MODE, instruction specific data, or any combination of the four (Instruction specific data is such information as shift direction, branch conditions, operation sizes, etc.). This provides you with over 1000 actual instructions, but keeps the total number of instructions small.

The MPU can perform add and subtract functions on BCD digits in groups of two digits. It can also multiply and divide binary numbers in these formats: 32-bit / 16-bit => 32-bit and 16-bit * 16-bit => 32-bit. The signed divide is the slowest instruction to execute on the 68000. It takes 170 clock cycles or 21.25 micrsec. at a clock speed of 8MHZ. This means that you could divide any 32-bit number in memory by any 16-bit number in memory over 47,000 times in ONE SECOND!!

I hope that this has given you a general idea of the power of the processor that the new Ataris will be using. If you are interested in more information, I would like to suggest a couple of books that are available. The first is called "68000 PRINCIPLES AND PROGRAMMING" by Leo Scanion. You should be able to order this book through the Howard Sams section of Compuserve (GO SAM). The second book I would like to recommend is "M68000 PROGRAMMERS REFERENCE MANUAL" available from:

Motorola Semiconductor Products Inc.
P.O. BOX 20912
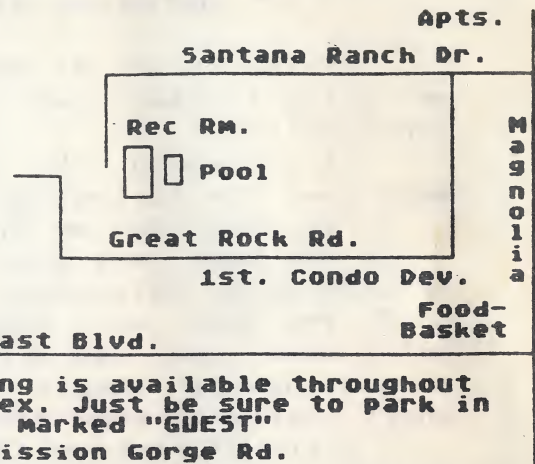Phoenix, AZ 85036   Ask for document # M68000UM(AD4).

ATR 8000 SIG

When: 7:00 pm
Forth Wednesday of each
Month
Where: Santana Ranch Estates
Recreation Room
(be sure to only park in
guest parking, please!)

San Diego Computer Society
When: 12:00 (swap meet)
1:30 (meeting)
Third Saturday of each
Month
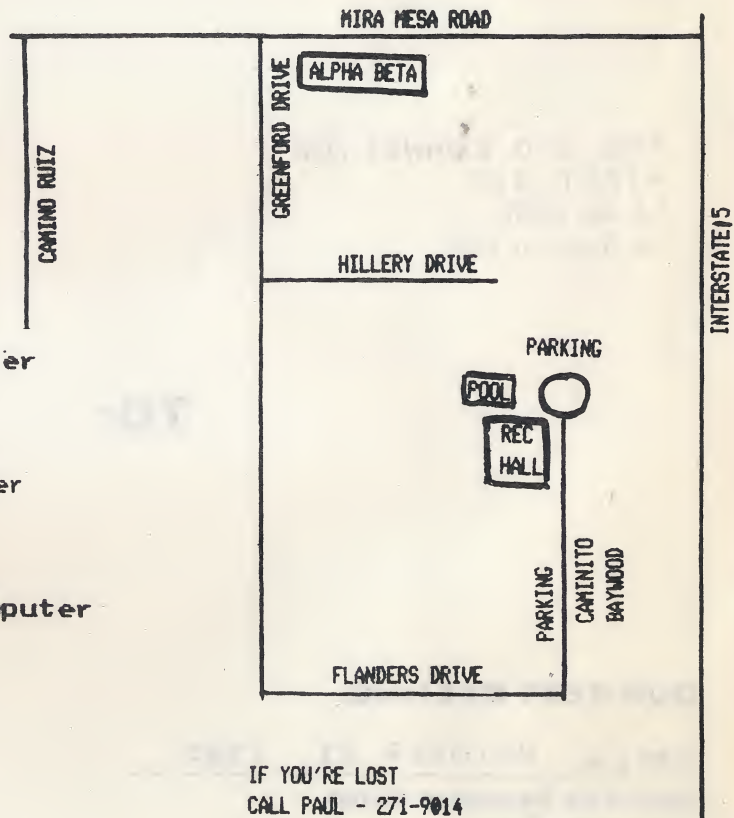Where: Mesa Collge, Appolo Theatre

ATR8000 SIG MEETING

```
              N
              △
                          Apts.
                 Santana Ranch Dr.
         ┌──────────────────┐      M
         │ Rec Rm.          │      a
         │ ┌─┐ ┌─┐ Pool      │      g
    ─────┘ │ │ │ │           │      n
           └─┘ └─┘           │      o
         │  Great Rock Rd.   │      l
         └──────────────────┘      i
             1st. Condo Dev.       a
                          Food-
        Mast Blvd.        Basket
```

Parking is available throughout
complex. Just be sure to park in
space marked "GUEST"
Mission Gorge Rd.

NORTH COUNTY MEETING PLACE

San Diego ATARI Computer Enthusiasts
When: 6:30 pm
Third Monday of each month
Where: North Park Recreation Center
4044 Idaho Street
Social Room

North County ATARI Computer Enthusiasts
When: 6:00 pm
Third Wednesday of each
month
Where: Caminito Baywood Rec. Hall
(see map)

```
                    MIRA MESA ROAD
        ┌─────────────┬───────────────────┐
        │          G  │ ┌──────────┐      │ I
        │          R  │ │ALPHA BETA│      │ N
        │          E  │ └──────────┘      │ T
      C │          E  │                   │ E
      A │          N  │                   │ R
      M │          F  │                   │ S
      I │          O  │                   │ T
      N │          R  │  HILLERY DRIVE    │ A
      O │          D  ├────────────       │ T
        │             │                   │ E
      R │          D  │         PARKING   │ /
      U │          R  │      ┌────┐  ○    │ 5
      I │          I  │      │POOL│        │
      Z │          V  │      └────┘        │
        │          E  │      ┌────┐        │
        │             │      │REC │   C P  │
        │             │      │HALL│   A A  │
        │             │      └────┘   M R  │
        │             │              I K   │
        │             │              N I   │
        │             │              I N   │
        │             │  FLANDERS    T G   │
        └─────────────┴──DRIVE──────O─────┘
                         BAYWOOD
```

IF YOU'RE LOST
CALL PAUL - 271-9014

ST SOFTWARE REVIEWERS WANTED BY ANTIC

As the first ST software starts arriving at Antic, we are going to need some reviewers -- particularly of business applications.

If you would like to review ST software for Antic, you must meet the following criteria:

1. You must own an ST.

2. You must have some area of special knowledge -- such as spreadsheets, databases, etc.

3. You must mail Antic a sample of your writing so that we'll know you can write reasonably well. (If you don't already have a sample of your reviewing style, write us a short review of some existing Atari product.)

Antic ST reviewers get paid $50 or more and may keep the software. Mail your writing sample and a letter explaining your areas of expertise to Jack Powell, Antic Magazine, 524 Second Street, San Francisco, CA 94107

THE I/O CONNECTOR
ATARI SIG
P.O. Box 203076
San Diego, Ca. 92120

TO:

OUR NEXT MEETING,

6:30 p.m. OCTOBER 21, 1985

North Park Recreation Center
4044 Idaho Street
Social Room